TD Théorie n°1

# *Indécidabilité par réduction*

## Introduction, Notations.

L'objectif de ce TD est de réaliser des réductions d'un problème à un problème indécidable. Dans ce TD, on ne traitera pas de réductions polynomiale à un problème **NP**-difficile.

Une machine $\mathscr{M}$ pourra être décrite sous forme d'un algorithme, d'un programme C, ou d'un programme OCaml. Quel que soit le format, on prendra soin de n'utiliser *que* des fonctions calculables, ce n'est pas toujours évident lorsqu'on écrit une machine sous forme d'algorithme.

Dans ce TD, la sérialisation d'un objet mathématique $x$ sera noté $\langle x \rangle$. Cette notation est là pour simplifier l'écriture des machines. Cependant, si une sérialisation n'est pas évidente pour un « type d'objet », on pourra préciser une fonction de sérialisation associée à ce « type ». Ainsi, on notera $\langle (x, y) \rangle$ la sérialisation du couple $(x, y)$ (ceci peut être étendu pour tout $n$-uplet). On notera $\langle \mathscr{M} \rangle$ la sérialisation de la machine $\mathscr{M}$ : dans le cas du coude C, ou du code OCaml, il s'agira du code source de $\mathscr{M}$, pour un algorithme, ce sera une description suffisamment précise de l'algorithme.

On notera $\mathscr{U}$ une machine universelle, calculant la fonction `interprète` :

$$\texttt{interprète}\,(\mathscr{M}, w) = \begin{cases} w' \text{ si } w \underset{\mathscr{M}}{\longrightarrow} w' \\ \text{non défini sinon} \end{cases}.$$

On pourra appeler cette fonction en C ou en OCaml avec la fonction `interprète` ou `exécute`. On utilisera le mot clé « Interprète » ou « Exécute » dans un algorithme.

On définit $w^{\text{rev}} = w_n w_{n-1} \ldots w_1$ le mot $w$ renversé où $w = w_1 w_2 \ldots w_n$. On notera $L^{\text{rev}} = \{w^{\text{rev}} \mid w \in L\}$ le langage $L$ renversé, et l'opération $L \mapsto L^{\text{rev}}$ est appelée *renversement*.

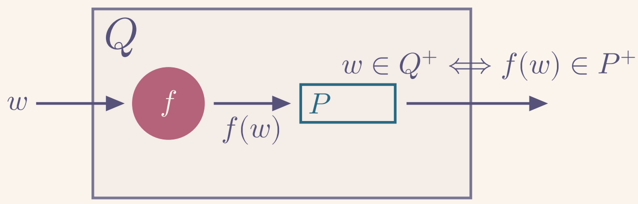
On notera $L_P$ le langage d'un problème $P$ (*i.e.* $P^+$). On définit le problème $\textsc{Appartient}_L$ qui décide si $w \in L$.

On rappelle la structure d'une réduction.

**Définition.** On dit qu'un problème $Q$ *se réduit à* $P$ dès lors qu'il existe une fonction $f : \mathscr{E}_Q \to \mathscr{E}_P$ *calculable* telle que :

$$w \in Q^+ \iff f(w) \in P^+.$$

On notera alors $Q \preccurlyeq P$.




Si $Q \preccurlyeq P$ et que $Q$ n'est pas décidable, alors $P$ le n'est pas non plus.

On rappelle le théorème de l'arrêt :

**Théorème.** Le problème $\textsc{Arrêt}$ est indécidable.

$\textsc{Arrêt}$ :
- **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine, et $w \in \Sigma^\star$
- **Sortie.** La machine $\mathscr{M}$ s'arrête-t-elle sur l'entrée $w$ ?

On pourra également utiliser les propriétés de stabilité des langages décidables par union, par concaténation, par intersection et par complémentaire, même si ça n'est pas le but premier de ce TD.

Par exemple, le problème CoArrêt (sur la page suivante) est aussi indécidable, c'est le complémentaire du problème Arrêt.

CoArrêt :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine, et $w \in \Sigma^\star$
**Sortie.** Est ce que la machine $\mathscr{M}$ ne s'arrête pas sur l'entrée $w$ ?

On admettra que les problèmes ci-dessous sont indécidables (ils viennent du TD 11). On pourra utiliser ces problèmes pour les réductions, mais pas ceux définis après la ligne.

**Parmi ces problèmes, un est décidable. Trouvez lequel !**

ArrêtUniv :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Est-ce que la machine $\mathscr{M}$ s'arrête sur toutes ses entrées ?

ArrêtExiste :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Est-ce que la machine $\mathscr{M}$ s'arrête sur une de ses entrées ?

ArrêtSimult :
**Entrée.** Deux sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de machines.
**Sortie.** Est-ce que les machines $\mathscr{M}$ et $\mathscr{N}$ s'arrêtent sur les mêmes entrées ?

Arrêt$_w$ :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** La machine $\mathscr{M}$ s'arrête-t-elle sur l'entrée $w$ ?

***Début du TD.***

Régulier :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il régulier ?

Fini :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il fini ?

Vide :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il vide ?

AccepteTout :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il $\Sigma^\star$ ?

CommencePar$_a$ :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Est-ce que la machine $\mathscr{M}$ s'arrête sur une entrée commençant par $a$ ?

EntréePair :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Est-ce que la machine $\mathscr{M}$ s'arrête sur une entrée de taille paire ?

LangagePair :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Est-ce que $\mathscr{L}(\mathscr{M})$ admet un nombre pair de mots ?

Local :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il local ?

LangageImpair :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Est-ce que $\mathscr{L}(\mathscr{M})$ admet un nombre impair de mots ?

Infini :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il infini ?

Palindrome :
**Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Les mots acceptés par $\mathscr{M}$ sont-ils tous des palindromes ?

RENVERSEMENT : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il stable par renversement[1] ?

CONCAT : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il la concaténation de deux langages décidables ?

DISJOINTS : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de deux machines.
**Sortie.** Les langages de $\mathscr{M}$ et $\mathscr{N}$ sont-ils disjoints ?

UNIONTOUT : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de deux machines.
**Sortie.** L'union des langages de $\mathscr{M}$ et $\mathscr{N}$ vaut-il $\Sigma^\star$ ?

CONCATTEST : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$, $\langle \mathscr{N} \rangle$ et $\langle \mathscr{K} \rangle$ de trois machines.
**Sortie.** Le langage de $\mathscr{M}$ est-il la concaténation du langage de $\mathscr{N}$ et de $\mathscr{K}$ ?

MAX3 : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Les mots acceptés par $\mathscr{M}$ sont-ils de taille inférieure ou égale à 3 ?

NONSTOP : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** La machine $\mathscr{M}$ ne s'arrête-t-elle pas, quel que soit l'entrée ?

MONOÏDE : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il un monoïde, pour la concaténation ?

OCAML : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** La machine $\mathscr{M}$ reconnaît-elle un programme OCAML ?

STABLECONCAT : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il stable par concaténation ?

TAILLEPREMIÈRE : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** La machine $\mathscr{M}$ accepte-t-elle les entrées dont la taille est un nombre premier ?

AUTOMATE : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il reconnu par un automate fini $\mathscr{A}$ ?

AUTOMATELOCAL : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il reconnu par un automate local $\mathscr{A}_{\text{loc}}$ ?

ACCEPTECOUPLE : **Entrée.** Les sérialisations $\langle \mathscr{M} \rangle$ et $\langle \mathscr{N} \rangle$ de deux machines, et deux mots $(u, v) \in (\Sigma^\star)^2$.
**Sortie.** La machine $\mathscr{M}$ accepte-t-elle $u$ si et seulement si $\mathscr{N}$ accepte $v$ ?

COMMUTATIF : **Entrée.** La sérialisation $\langle \mathscr{M} \rangle$ d'une machine.
**Sortie.** Le langage de $\mathscr{M}$ est-il commutatif,[2] pour la concaténation ?

***Fin du TD.***

[1]Un langage $L$ est stable par renversement si $L^{\text{rev}} = L$.
[2]Un langage $L$ est commutatif si, pour $w = uv \in L$, on a $vu \in L$, quel que soit la découpe de $w$.